

Enabling new capabilities and insights from quantum chemistry by using component architectures

C L Janssen¹, J P Kenny¹, I M B Nielsen¹, M Krishnan², V Gurumoorthi², E F Valeev³, T L Windus²

¹ Sandia National Laboratories, P.O. Box 969, Livermore, CA 94551-0969

² Pacific Northwest National Laboratory, P.O. Box 999, Richland, WA 99352

³ Virginia Polytechnic Institute and State University Blacksburg, VA 24061

E-mail: cljanss@sandia.gov

Abstract.

Steady performance gains in computing power, as well as improvements in scientific computing algorithms, are making possible the study of coupled physical phenomena of great extent and complexity. The software required for such studies is also very complex and requires contributions from experts in multiple disciplines. We have investigated the use of the Common Component Architecture (CCA) as a mechanism to tackle some of the resulting software engineering challenges in quantum chemistry, focusing on three specific application areas. In our first application, we have developed interfaces permitting solvers and quantum chemistry packages to be readily exchanged. This enables our quantum chemistry packages to be used with alternative solvers developed by specialists, remedying deficiencies we discovered in the native solvers provided in each of the quantum chemistry packages. The second application involves development of a set of components designed to improve utilization of parallel machines by allowing multiple components to execute concurrently on subsets of the available processors. This was found to give substantial improvements in parallel scalability. Our final application is a set of components permitting different quantum chemistry packages to interchange intermediate data. These components enabled the investigation of promising new methods for obtaining accurate thermochemical data for reactions involving heavy elements.

1. Introduction

Quantum chemistry is concerned with obtaining accurate properties for atoms, molecules, and larger complexes and structures from first principles (*ab initio*). The primary tool for doing this is the Schrödinger equation:

$$H = -\frac{1}{2} \sum_i^n \nabla_i^2 - \sum_i^n \sum_a^N \frac{q_a}{r_{ia}} + \sum_{i>j}^n \frac{1}{r_{ij}} + \sum_{a>b}^N \frac{q_a q_b}{r_{ab}} \quad (1)$$

where i and j refer to each of the n electrons, and a and b refer to each of the N nuclei. As written here, each of the nuclear coordinates is treated as a fixed parameter, and the quantum mechanical equations for the nuclei are solved separately (the Born-Oppenheimer approximation). Beginning at such a fundamental starting point, quantum chemistry is not tuned for any particular problem and can be applied to a wide range of areas, including thermochemistry (for example, for optimization of industrial processes), catalysis, structure

and function in biology, nanotechnology, and, indeed, any area where understanding chemical properties is important.

The Schrödinger equation is quite complex to solve, and, consequently, many approximate methods have been developed to solve it. In quantum chemistry, the method to be applied must be selected according to the property of interest, the size and nature of the system, and the accuracy needed. Due to constraints in the available quantum chemistry program suites and the accessible computational resources, it might be necessary to compromise by selecting a smaller system that mimics the system of interest or to settle for less accuracy. Often the quantum chemical method is supplemented with more approximate methods. Incorporating force field methods, for example, yields hybrid quantum mechanical/molecular mechanical (QM/MM) methods, in which a subsystem of atoms requiring a high-accuracy treatment (for example, a reactive center in an enzyme) are treated quantum mechanically and are coupled to a larger set of atoms treated classically. Increases in available computational power and algorithmic sophistication will increase the sizes of systems that can be treated. As a result, QM/MM and other multi-scale and multi-physics applications are going to constitute a larger portion of future quantum chemistry applications. Programming challenges posed by this development include providing a framework for allowing the combination of quantum chemistry codes with codes developed in other disciplines.

A wide range of methods are available in quantum chemistry, including density functional theory, perturbation theory, coupled-cluster theory, configuration interaction, and multi-reference methods, each of which has many variants. A thorough, accurate study of a given chemical system will involve several of the methods mentioned above, but because of the plethora of algorithm variations, no single quantum chemistry program suite implements all of these methods. Furthermore, many potential combinations of techniques are not available as a consequence of the limited abilities to interface software suites. Additionally, a quantum chemistry study may involve computation of corrections that require modifications to the Schrödinger equation, such as relativistic effects and Born-Oppenheimer corrections. While some of the correction terms are, in principle, simple to incorporate into any method, it is difficult in practice to share internal data between different computational chemistry codes, frustrating attempts to make such corrections more widely available. Development of interfaces allowing more straightforward data sharing between quantum chemistry programs and promoting integration of different program suites is therefore an important prerequisite for leveraging the combined strengths and capabilities of multiple codes.

In the present work, we address some of the programming challenges outlined above through the use of component architectures. We will briefly summarize component architectures and then discuss three applications of the component approach to quantum chemistry:

- the implementation of high-level components that enable one quantum chemistry method and program suite to be easily substituted for another,
- the use of components to better utilize computing resources,
- and the use of low-level components to share data between program suites, enabling creation of new functionalities by combined use of capabilities from different program packages.

2. Overview of Component Architectures

Within computational sciences, component approaches are showing great utility as the foundations of approaches to collaborative software development [1–4]. While the concept of software components is difficult to define [5], the fundamental distinction between the library and component approaches to code construction is that components exist in a runtime environment, which is provided by a *framework*. Based on the interfaces shared between components, a component framework dynamically composes applications without code relinking, yielding an

environment for plug and play application construction. This runtime configuration may be tightly controlled by application programmers or delegated to the users, providing sophisticated users with a great degree of flexibility. Users may, in principle, construct applications that were not even envisioned by the component developers.

The Common Component Architecture (CCA) has evolved to bring component-based software engineering to scientific computing, providing the performance required by demanding science applications, while respecting parallel computing models [6]. The full power of the CCA approach is realized with integration of the Babel code interoperability tool [7, 8], which allows various programming languages to coexist in the same application, includes a language-neutral interface definition language, which aids interface design, and implements a uniform object-oriented programming model and scientific data types for all supported languages. The components described in this work conform to the CCA specification, including Babel language interoperability, and were used within the Ccaffeine [9] CCA-compliant framework.

Following the CCA/Babel approach, the cooperating development groups created shared interface definitions using Babel’s Scientific Interface Definition Language (SIDL). These interfaces form so-called ports, which a CCA framework connects during application composition. For components providing high-level functionality, such as the optimization application described in Section 3, the implementations of these interfaces are primarily simple wrappers for existing codes. When lower-level functionality is exposed via component technology, as in the molecular integral sharing described in Section 5, various implementation details are exposed, creating demands not only for interface specification but also for standardization of conventions and data layouts. The benefits of low-level data sharing for code development and the performance costs due to data reformatting are active areas of the authors’ research.

As a development paradigm providing a domain-agnostic runtime environment and language interoperability solution, and as a catalyst for interface design and the standardization of conventions for low-level data, the CCA approach to software development enables large-scale, collaborative software development. By implementation of both high- and low-level code interfaces between multiple development groups, domains, and programming languages, the following sections demonstrate the utility of component approaches for scientific computing.

3. High-Level Quantum Chemistry Components

Quantum chemistry packages currently tend to be stand-alone program suites. While this is convenient for users and allows developers to maintain complete control over their packages, it results in much replicated effort. This is true not only between various quantum chemistry packages but across application domains that require similar underlying functionality. Our initial CCA application to quantum chemistry allowed our two packages, the Massively Parallel Quantum Chemistry program (MPQC) [10] and NWChem [11], to utilize the same set of optimization components. The optimization components were written by optimization experts and implemented using the Toolkit for Advanced Optimization (TAO) [12]. We jointly developed generic interfaces, implemented them in each of our software packages, and used the interfaces in optimization applications. The resulting architecture is shown in Figure 1. Each of the components in this figure is fully described in Reference 3.

Using easily interchangeable software components, we were able to compare a variety of methods for solving the same problem. Each of the solvers in NWChem, MPQC, and TAO employed a variant of the quasi-Newton approach. Table 1 shows the number of gradient and function evaluations for the optimization of several molecules using both the stand-alone NWChem optimizer and the TAO CCA components. The stand-alone NWChem optimizer was employed both with and without line searches enabled. The results indicate that line searches can often reduce the number of gradient evaluations needed, but the large number of energy

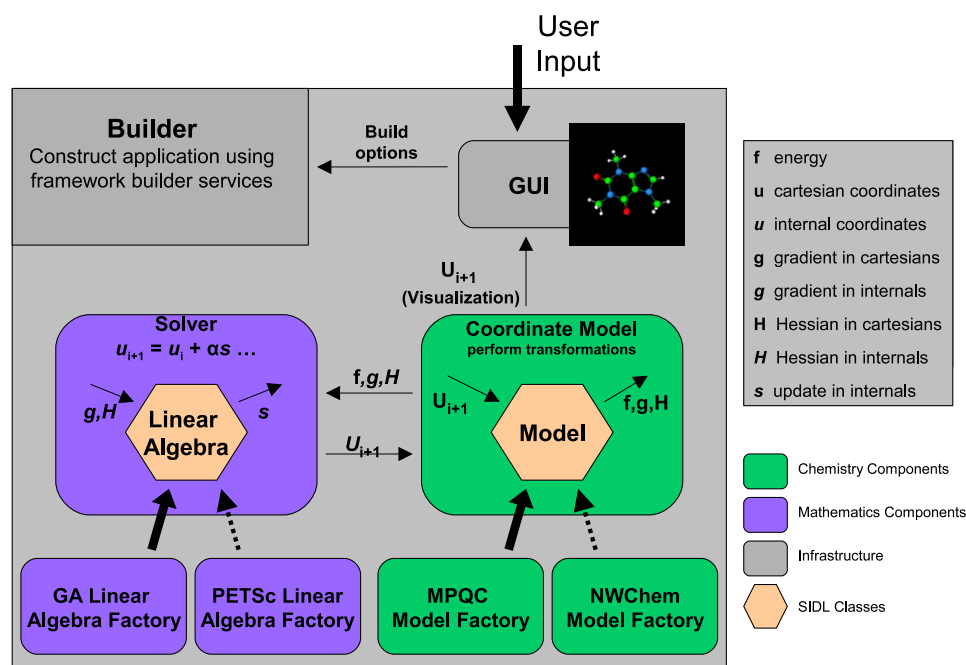


Figure 1. A schematic depiction of the optimization component architecture.

points causes the execution time to increase. The TAO package uses gradient information with cubic interpolation to adjust the step length in the line search, performing significantly better in several cases. Similarly, the TAO optimizer sometimes perform substantially better than MPQC's stand-alone optimizer [3]. These results have provided much useful information to the MPQC and NWChem code teams and produced ideas for further collaboration between the TAO developers and the quantum chemists. Since this optimization work was published, these same high-level interfaces have been implemented within the GAMESS [13] computational chemistry code. This addition broadens the computational methods available through the the CCA framework and lays the groundwork for further integration between multiple codes.

4. Components for Better Resource Utilization

The single program/multiple data (SPMD) model is the most commonly used technique for parallel applications. The analogous component-based technique constructs applications from components, each of which uses the distributed resources of the entire machine when executing, resulting in a single component/multiple data (SCMD) model. Unfortunately, this is not always the most efficient way to use the machine. While programmers strive to write efficient algorithms that give good scaling as the node count and the problem size increase, problems are encountered for which more processors are available than can be efficiently utilized. However, by dividing the work into multiple, independent tasks, each using a subset of the processors that is small enough to obtain good parallel efficiency, the overall parallel efficiency can be improved. Some quantum

Table 1. The number of energy and gradient computations needed for molecular structure optimizations in Cartesian coordinates at the HF/6-31G level of theory. The stand-alone optimizer for NWChem both with and without line search was used, as well as the TAO optimizer through the component-based interface.

Molecule	NWChem (no line search)		NWChem (line search)		NWChem/TAO	
	Energy	Gradient	Energy	Gradient	Energy	Gradient
glycine	33	33	65	33	19	19
isoprene	56	56	89	45	45	45
phosphoserine	79	79	121	61	67	67
acetylsalicylic acid	43	43	83	42	51	51
cholesterol	33	33	>194	>98	30	30

chemistry applications are well suited to decomposition in this manner. For example, when numerically evaluating the gradient or Hessian of the molecular energy with respect to changes in the nuclear coordinates, energies and gradients must be computed at numerous geometries displaced slightly from the equilibrium geometry. Each of these computations can be run on a subset of the processors, independently of the computations at other displacements. The trivial case of running each displacement on a single node is not sufficient, since the computations may be too large to run on one node. Thus, it is not enough to simply ignore parallelism in the individual energy computations.

A multiple component/multiple data (MCMD) approach, based on the CCA, has been implemented using NWChem to perform Hessian evaluations in this manner [2]. To accomplish this, the Global Array library was modified to permit data allocations on subsets of nodes, as specified by the programmer. The CCA driver component, which had responsibility for the overall computation, instantiated several NWChem components for gradient computations on disjoint subsets of nodes. Each of these NWChem gradient components then performed multiple energy computations in parallel on its subset of nodes. Thus, two levels of parallelism were employed. In an application of this approach for a simple cluster of five water molecules, an order of magnitude improvement in time to solution relative to the SCMD approach was observed for 256 processors [2].

Multiple chemistry applications would benefit from this model, including properties along a reaction path, vibrational self consistent field, various Monte Carlo simulations, and constrained optimizations along a linear synchronous transit path. In fact, as chemists start to examine very complex molecular systems that require many potential energy computations to determine kinetic and thermodynamic properties, this model will become much more common. The MCMD model is also applicable to other areas of science such as those where sensitivity analysis, parameter studies, and statistical simulations are required.

5. Low-Level Quantum Chemistry Components

Ab initio quantum chemical studies of molecular systems often entail application of many different levels of theory, including different correlation methods and basis sets, as well as the computation of various corrections such as relativistic and core-valence correlation effects. To carry out all the computations required for a given study, it is frequently necessary to apply several different *ab initio* quantum chemistry program packages because available packages each include a limited range of methods or because the optimum implementations for the various

methods of interest are found in different codes.

Application of several *ab initio* program suites in a particular study is cumbersome, requiring familiarity with the use of several programs, necessitating setting up customized inputs for each individual package, providing basis sets in different formats, etc. An intriguing opportunity provided by the CCA, however, is the possibility of sharing data at very low levels, thereby permitting a quantum chemistry package to directly leverage the strengths of other packages to deliver new functionality and increase the range of scientific studies that can be accomplished.

To this end, we have recently developed interfaces to the one- and two-electron integral computations that are central to quantum chemistry applications. A variety of integral types are useful, and no single package implements all of them. For example, MPQC does not implement the integrals needed for the Douglas-Kroll approximation [14] for computing the scalar relativistic effect, while NWChem and GAMESS [13] do. On the other hand, MPQC implements (via the Libint [15] package) the two-electron integrals needed for explicitly correlated methods, which NWChem and GAMESS lack. The ability to access implementations of both of these types of integrals from a single package makes possible the exploration of new all-electron, explicitly correlated methods for describing molecular systems containing heavy elements. Specifically, using these CCA integral interfaces, we have implemented the Douglas-Kroll-Hess [16] relativistic correction in conjunction with explicitly correlated second-order Møller-Plesset perturbation theory MP2-R12 [17–19] (using the MP2-R12/A variant, which employs the standard approximation A). The MP2-R12 method is a variant of MP2 theory that incorporates terms linear in the inter-electronic distance r_{12} into the wave function, making the method capable of efficiently producing energies close to the complete basis limit. In this study, an auxiliary basis set (ABS) was used for the resolution of the identity required in MP2-R12 [20,21], and the complementary ABS (CABS) formalism [22] was employed.

As a first application of this new capability, we have carried out *ab initio* computations that were not previously possible for the chromium compounds $\text{Cr}(\text{OH})_6$ and $\text{CrO}(\text{OH})_4$. Corrosion processes involving these species appear in many industrial applications, including chemical vapor deposition employed in the semi-conductor industry, and modeling of these processes requires accurate thermochemical data for the species involved. In a previous study we computed heats of formation for these species using the coupled-cluster method CCSD(T), and we applied corrections for basis set incompleteness, core-valence correlation effects, and scalar relativistic effects computed at the MP2 level. These corrections were all computed separately, assuming they were additive, and their computation involved invoking three different *ab initio* quantum chemistry packages. Heats of formation for $\text{Cr}(\text{OH})_6$ and $\text{CrO}(\text{OH})_4$ were obtained by computing reaction energies for the isogyric reactions



and using literature values for the heats of formation for CrO_3 and H_2O .¹

We have now obtained new reaction energies for the above reactions, applying corrections for basis set incompleteness, core-valence correlation effects, and scalar relativistic effects computed at the auxiliary basis set MP2-R12/A level of theory. The orbital basis employed was the correlation-consistent basis set aug-cc-pwCVTZ-DK [24] for Cr and the aug-cc-pVTZ-DK set [25] for O, H, and the auxiliary basis sets were completely decontracted versions of the aug-cc-pwCVQZ-DK set for Cr and the aug-cc-pvQZ-DK set for O, H. The decontracted orbital basis sets were used for the p^2 basis in the relativistic calculations. All three corrections were computed in a coupled fashion by doing a single Douglas-Kroll-Hess computation at the MP2-R12/A level,

¹ See Reference 23 for details.

Table 2. Evaluation of $\Delta H_{f,0}^\circ$ and $\Delta H_{f,298.15}^\circ$ for $\text{Cr}(\text{OH})_6$ and $\text{CrO}(\text{OH})_4$. All entries are in kcal mol^{-1} .

	$\text{Cr}(\text{OH})_6$ Reaction 1		$\text{CrO}(\text{OH})_4$ Reaction 2	
$\Delta E_{\text{rxn}}[\text{HF}]$	−29.66		23.58	
$\delta[\text{MP2}]$	+9.81		−7.47	
$\delta[\text{CCSD}]$	+5.04		+8.45	
$\delta[\text{CCSD}(\text{T})]$	+9.94		+0.96	
	MP2	MP2-R12	MP2	MP2-R12
$\delta[\text{basis}]$	−3.55		−2.39	
$\delta[\text{core}]$	−0.32		−0.79	
$\delta[\text{rel}]$	+2.08		+3.44	
$\delta[\text{basis+core+rel}]$		−1.85		+0.32
$\delta[\text{ZPVE}]$	−8.07	−8.07	−4.99	−4.99
$\Delta H_{\text{rxn},0}^\circ$	−14.73	−14.79	20.79	20.85
$\Delta H_{f,0}^\circ$	−232.98	−232.92	−211.40	−211.45
$\Delta H_{f,298.15}^\circ$	−239.77	−239.71	−215.94	−216.00

including the Cr 3s3p orbitals in the correlation procedure. Our results are summarized in Table 2. The table lists the computed reaction energies for Reactions 1 and 2 and the resulting heats of formation for $\text{Cr}(\text{OH})_6$ and $\text{CrO}(\text{OH})_4$. In the table, $\Delta E_{\text{rxn}}[\text{HF}]$ is the computed reaction energy at the Hartree-Fock level using the cc-pVTZ basis set, and $\delta[\text{MP2}]$, $\delta[\text{CCSD}]$, and $\delta[\text{CCSD}(\text{T})]$ represent the increment in the reaction energy relative to the preceding level of theory. The quantities $\delta[\text{basis}]$, $\delta[\text{core}]$, $\delta[\text{rel}]$, and $\delta[\text{ZPVE}]$ denote the contributions to the reaction energy from basis set improvement, core-valence correlation, scalar relativistic effects, and zero-point vibrational energy. For comparison we have also listed the values of the corrections when computed individually with conventional MP2 theory using the basis sets aug-cc-pV5Z, aug-cc-pwCVTZ, and aug-cc-pVTZ-DK for computing the basis set correction, core-valence correlation correction, and scalar-relativistic correction, respectively.

The MP2-R12 computation of these corrections yields values of −1.85 and 0.33 kcal mol^{-1} for Reactions 1 and 2, respectively. The individually computed MP2 corrections for these reactions add up to −1.79 and 0.26 kcal mol^{-1} , respectively, and the two approaches, thus, yield very similar results. This similarity suggests two conclusions: that MP2-R12 can, even with a limited orbital basis set, account for the effect of $\delta[\text{basis}]$ and that our previous approach of separately computing $\delta[\text{basis}]$, $\delta[\text{core}]$, and $\delta[\text{rel}]$ was valid. However, we believe that the computation of all three effects in a coupled manner is, in general, a superior approach, and MP2-R12 allows us to do this efficiently. The new computation of the heats of formation of $\text{Cr}(\text{OH})_6$ and $\text{CrO}(\text{OH})_4$ results in a revision of the previously computed values [23] by 2–3 kcal mol^{-1} and yields the values $\Delta H_{f,0}^\circ[\text{Cr}(\text{OH})_6] = -233$, $\Delta H_{f,298.15}^\circ[\text{Cr}(\text{OH})_6] = -240$, $\Delta H_{f,0}^\circ[\text{CrO}(\text{OH})_4] = -211$, and $\Delta H_{f,298.15}^\circ[\text{CrO}(\text{OH})_4] = -216$ kcal mol^{-1} .

Finally, we note that application of Douglas-Kroll corrections to MP2-R12 theory is not as straightforward as for other methods. Two-electron integrals of the commutator and double commutator of the Fock operator, \hat{f} , and the correlation factor, in this case r_{12} , arise in these

methods:

$$\int d\bar{r}_1 d\bar{r}_2 \phi_1(\bar{r}_1) \phi_2(\bar{r}_1) [\hat{f}, r_{12}] \phi_3(\bar{r}_2) \phi_4(\bar{r}_2) \quad (2)$$

$$\int d\bar{r}_1 d\bar{r}_2 \phi_1(\bar{r}_1) \phi_2(\bar{r}_1) [r_{12}, [\hat{f}, r_{12}]] \phi_3(\bar{r}_2) \phi_4(\bar{r}_2) \quad (3)$$

In the nonrelativistic case, the only components of the Fock operator that do not commute with r_{12} are the kinetic energy, $\hat{T}_i = -\nabla_i^2/2$, and the exchange operator, \hat{K} . The single and double commutators with the kinetic energy operator are usually evaluated analytically, whereas the exchange commutators are either evaluated by inserting the resolution of the identity (standard approximation B) or dropped altogether (standard approximation A). Similar strategies can be pursued for the relativistic case. In our current application of the MP2-R12/A method, we approximate the commutators of the Fock operator by their nonrelativistic kinetic energy counterparts; however, we use the Fock matrix elements and orbitals computed relativistically. The validity of the current approach will be tested in the future by benchmarking against more rigorous methods, MP2-R12/B and MP2-R12/C. Their nonrelativistic versions have been implemented in MPQC and are currently being extended to the relativistic regime.

6. Conclusions

We have reviewed some of our work to investigate the use of the Common Component Architecture (CCA) as a mechanism to overcome some of the software engineering difficulties that arise in modern quantum chemistry. We have presented three applications, each of which exercises the CCA in different ways. In our first application, we developed components that permit easy exchange of both solvers and quantum chemistry packages. This made it possible for our quantum chemistry packages to use carefully designed solvers developed by specialists instead of the native solvers that are provided with each quantum chemistry package. As a result, geometry optimizations could be performed more efficiently, requiring shorter time to solution. In the second application, a multiple component/multiple data (MCMD) programming model was developed, permitting concurrent execution of multiple components, each of which is parallel. This permitted hierarchical parallel computation to be employed in a way that can improve parallel efficiency in certain cases. In one such case, an order of magnitude improvement was observed in total computation time. The final application involved devising a set of low-level components that enables the one- and two-electron integrals computed by one quantum chemistry package to be used by another package. This capability was used to develop a new method combining relativistic effects and an explicitly correlated electronic structure method. In a preliminary application of this method, we computed thermochemical data for chromium hydroxides, providing the most accurate heats of formation available for these compounds to date.

Acknowledgments

Funding for this work was provided by the United States Department of Energy's Scientific Discovery through Advanced Computing program.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

This research was performed in part using the Molecular Science Computing Facility in the William R. Wiley Environmental Molecular Sciences Laboratory, a national scientific user facility sponsored by the U.S. Department of Energy, Office of Biological Environmental Research and located at the Pacific Northwest National Laboratory operated for the DOE by Battelle.

References

- [1] L. Curfman McInnes, B. A. Allan, R. Armstrong, S. J. Benson, D. E. Bernholdt, T. L. Dahlgren, L. Freitag Diachin, M. Krishnan, J. A. Kohl, J. W. Larson, S. Lefantzi, J. Nieplocha, B. Norris, S. G. Parker, J. Ray, and S. Zhou. *Solutions of PDEs on Parallel Computers*, chapter Parallel PDE-Based Simulations Using the Common Component Architecture. Springer-Verlag, New York, 2005.
- [2] M. Krishnan, Y. Alexeev, T. L. Windus, and J. Nieplocha. Multilevel parallelism in computational chemistry using common component architecture and global arrays. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, Santa Fe, 2005. IEEE Press.
- [3] J. P. Kenny, S. J. Benson, Y. Alexeev, J. Sarich, C. L. Janssen, L. Curfman McInnes, M. Krishnan, J. Nieplocha, E. Jurrus, Carl Fahlstrom, and T. L. Windus. *J Comput Chem*, 25:1717–1725, 2004.
- [4] S. Lefantzi, J. Ray, and H. N. Najm. Using the common component architecture to design high performance scientific simulation codes. In *Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS 2003)*, Santa Fe, 2003. IEEE Press.
- [5] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley, New York, 2002.
- [6] David E. Bernholdt, Benjamin A. Allan, Robert Armstrong, Felipe Bertrand, Kenneth Chiu, Tamara L. Dahlgren, Kostadin Damevski, Wael R. Elwasif, Thomas G. W. Epperly, Madhusudhan Govindaraju, Daniel S. Katz, James A. Kohl, Manoj Krishnan, Gary Kumfert, J. Walter Larson, Sophia Lefantzi, Michael J. Lewis, Allen D. Malony, Lois C. McInnes, Jarek Nieplocha, Boyana Norris, Steven G. Parker, Jaideep Ray, Sameer Shende, Theresa L. Windus, and Shujia Zhou. A component architecture for high-performance scientific computing. *Intl J High-Perf Computing Appl*, 20:163–202, 2006.
- [7] T. Dahlgren, T. Epperly, and G. Kumfert. Babel user’s guide, January 2004.
- [8] Babel homepage. <http://www.llnl.gov/CASC/components/babel.html>.
- [9] B. A. Allan, R. C. Armstrong, A. P. Wolfe, J. Ray, D. E. Bernholdt, and J. A. Kohl. The CCA core specification in a distributed memory SPMD framework. *Concurrency and Computation: Practice and Experience*, 14(5):1–23, 2002.
- [10] Curtis L. Janssen, Ida B. Nielsen, Matt L. Leininger, Edward F. Valeev, Joseph P. Kenny, and Edward T. Seidl. The Massively Parallel Quantum Chemistry Program (MPQC), Version 2.4.0 Prerelease. <http://www.mpqc.org>.
- [11] R. A. Kendall, E. Apra, D. E. Bernholdt, E. J. Bylaska, M. Dupuis, G. I. Fann, R. J. Harrison, J. L. Ju, J. A. Nichols, J. Nieplocha, T. P. Straatsma, T. L. Windus, and A. T. Wong. High performance computational chemistry: An overview of NWChem, a distributed parallel application. *Comput Phys Commun*, 128:260–270, 2000.
- [12] Steve Benson, Lois Curfman McInnes, Jorge Moré, and Jason Sarich. TAO user’s manual. Technical Report ANL/MCS-TM-242, Mathematics and Computer Science Division, Argonne National Laboratory, 2004. See <http://www.mcs.anl.gov/tao>.
- [13] General Atomic and Molecular Electronic Structure System (GAMESS). <http://www.msg.fci.ameslab.gov/GAMESS/>.
- [14] M. Douglas and N. M. Kroll. Quantum electrodynamical corrections to the fine structure of helium. *Ann Phys*, 82:89–155, 1974.
- [15] Libint homepage. <http://www.ccmst.gatech.edu/evaleev/libint/>.
- [16] B. A. Hess. Relativistic electronic-structure calculations employing a two-component no-pair formalism with external-field projection operators. *Phys Rev A*, 33:3742–3748, 1986.
- [17] W. Kutzelnigg. R12-dependent terms in the wave-function as closed sums of partial-wave amplitudes for large- l . *Theor. Chim. Acta*, 68:445–469, 1985.
- [18] W. Kutzelnigg and W. Klopper. Wave functions with terms linear in the interelectronic coordinates to take care of the correlation cusp. I. General theory. *J. Chem. Phys.*, 94:1985–2001, 1991.
- [19] W. Klopper. chapter r_{12} -Dependent Wavefunctions. John Wiley & Sons, Chichester, UK, 1998.
- [20] W. Klopper and C. C. M. Samson. Explicitly correlated second-order Møller-plesset methods with auxiliary basis sets. *J. Chem. Phys.*, 116:6397–6410, 2002.
- [21] E. F. Valeev and C. L. Janssen. Second-order Moller-Plesset theory with linear R12 terms (MP2-R12) revisited: Auxiliary basis set method and massively parallel implementation. *J Chem Phys*, 121:1214–1227, 2004.
- [22] E. F. Valeev. Improving on the resolution of the identity in linear R12 ab initio theories. *Chem. Phys. Lett.*, 395:190–195, 2004.
- [23] I. M. B. Nielsen and M. D. Allendorf. Thermochemistry of the chromium hydroxides $\text{Cr}(\text{OH})_n$, $n=2-6$, and the oxyhydroxide $\text{CrO}(\text{OH})_4$: Ab initio predictions. *J. Phys. Chem. A*, 110:4093, 2006.
- [24] N. B. Balabanov and K. A. Peterson. *J. Chem. Phys.*, 123:064107, 2005.
- [25] T. H. Dunning, Jr. *J. Chem. Phys.*, 90:1007, 1989.